Le développement d'une plate-forme pour l'annotation spécialisée de documents Web : retour d'expérience

Thierry Hamon, Adeline Nazarenko

Le développement d'une plate-forme pour l'annotation spécialisée de documents Web : retour d'expérience

Thierry Hamon — Adeline Nazarenko

LIPN – UMR 7030, Université Paris 13 - CNRS
99, av. J-B Clément - F-93430 Villetaneuse
{thierry.hamon, adeline.nazarenko}@lipn.univ-paris13.fr

RÉSUMÉ. *Au-delà des moteurs de recherche généralistes, des outils capables d'interroger des collections documentaires spécialisées doivent être proposés pour répondre à des besoins d'information précis. Cela suppose une analyse sémantique adaptée à la collection de documents et au domaine considérés. C'est l'objectif de la plate-forme d'annotation Ogmios décrite ici. Des solutions sont proposées pour résoudre les contraintes opérationnelles et les problèmes d'interopérabilité spécifiques à ce cadre applicatif : la distribution les traitements, l'encapsulation des outils de TAL intégrés, et la définition d'une architecture unique pour l'annotation de gros volumes de documents et la construction de ressources spécialisées à partir d'un corpus d'acquisition. Les performances obtenues pour l'annotation des documents issus du Web sont compatibles avec le rythme de leur récupération. Nous montrons également comment cette plate-forme a été intégrée dans un moteur de recherche spécialisé.*

ABSTRACT. *Beyond general search engines, tools able to mine specialised document collections are required to answer precise queries. A semantic analysis adapted to the documents and the domain must be previously performed. This is the role of the Ogmios platform. Operational constraints and interoperability problems have been solved by distributing the annotation process, wrapping NLP tools into modules, and integrating in a common architecture the annotation of a collection of documents and the building of semantic resources from acquisition corpora. The performance for the annotation of web documents is compatible with the speed of the document crawling. We also explain how the platform has been integrated in a specialised search engine.*

MOTS-CLÉS : *interopérabilité, format d'annotation, robustesse, annotation linguistique, corpus spécialisés, acquisition de connaissances, moteur de recherche spécialisé.*

KEYWORDS: *interoperability, annotation format, robust processing, linguistic annotation, specialised corpora, knowledge acquisition, specialised search engine.*

1. Introduction

Même si le traitement automatique des langues (TAL) a fait de considérables progrès en termes de performances avec l'éclosion de besoins massifs de traitement de corpus, réaliser une analyse linguistique de documents issus du Web demeure aujourd'hui une gageure. D'un côté les performances des outils de TAL s'accordent mal avec l'hétérogénéité caractéristique du Web et le volume de données qu'il comporte, d'un autre côté, le TAL est important pour développer des méthodes fines d'accès au contenu des documents issus du Web car il existe des besoins d'information spécialisée que les moteurs de recherche grand public, outils généralistes par excellence, ne peuvent satisfaire.

C'est pour répondre à ce type de besoins et effectuer un traitement spécialisé de collections de documents issus du Web que nous avons développé Ogmios, plate-forme d'annotation destinée à enrichir des documents de diverses informations linguistiques et sémantiques. Elle a été conçue initialement, dans le cadre du projet ALVIS, pour des moteurs de recherche spécialisés, mais elle peut être intégrée dans des applications variées. Cette plate-forme est actuellement disponible[1].

Dans cet article, nous faisons le bilan du travail de développement d'Ogmios. La partie 2 situe le contexte de ce travail et ses objectifs. Réaliser une analyse spécialisée et donc sémantique des documents tout en tenant compte de la volumétrie et de l'hétérogénéité caractéristiques des données issus du Web était un défi. La partie 3 analyse les difficultés auxquelles il a fallu faire face et les solutions que nous avons proposées. Les parties 4 et 5 présentent la plate-forme Ogmios en elle-même et certaines expériences dans lesquelles elle a été utilisée. La partie 6 tente de faire le bilan de cette expérience en soulignant les défis qui restent à relever.

2. Contexte et objectifs

2.1. *Vers des moteurs de recherche sémantiques spécialisés*

Au-delà des moteurs de recherche grand public, il existe des moteurs spécialisés comme celui de la *National Library of Medecine*, PubMed, qui repose sur un intense travail documentaire et qui donne accès à l'information scientifique en médecine et en biologie. Il est évident qu'un biologiste qui s'intéresse à une espèce animale particulière va interroger PubMed[2], plutôt qu'un moteur généraliste. Pourtant, PubMed ne répond souvent qu'imparfaitement à ses besoins : le champ scientifique couvert restant très large, de nombreux termes ambigus viennent brouiller les résultats (*transfert* n'a pas le même sens en génomique et en psychologie) ; des connaissances précises sur le sens des mots sont nécessaires pour interpréter les requêtes de manière pertinente

1. Sous la forme d'un module CPAN (`http://search.cpan.org/~thhamon/Alvis-NLPPlatform/`).
2. `http://www.ncbi.nlm.nih.gov/PubMed/`

et fournir par exemple des documents parlant de *choc thermique* en réponse à une requête sur les *facteurs de stress* moléculaires[3]. Il faut donc pouvoir construire à façon des moteurs diversement spécialisés pour répondre aux besoins particuliers de telle ou telle communauté d'utilisateurs.

C'était l'un des objectifs du projet ALVIS (*Superpeer semantic Search Engine*)[4]. Ce projet visait à développer des moteurs de recherche *open source* sémantiques capables de prendre en compte à la fois le thème et le contexte de la recherche pour affiner l'analyse de la requête et du document. Il s'agissait par ailleurs d'intégrer ces différents moteurs de recherche dans une architecture *peer-to-peer*. Le système global est constitué d'un réseau de « nœuds » moteurs de recherche, chacun pouvant être spécialisé dans un domaine particulier. Les nœuds spécialisés proposent une véritable analyse du contenu textuel pour améliorer l'accès au document. Étant donné la puisssance de calcul nécessaire, un nœud peut être implanté sur plusieurs machines, les traitements nécessaires à un moteur de recherche particulier pouvant eux-mêmes être effectués en parallèle. L'objectif du projet consistait donc à développer, outre l'architecture P2P, les outils permettant l'émergence de moteurs de recherche spécialisés, c'est-à-dire l'ensemble des briques logicielles permettant à un utilisateur ou à une communauté d'utilisateurs de construire un moteur de recherche spécialisé sur un sujet donné et doté de fonctionnalités sémantiques avancées. Les briques élémentaires sont le moissonneur spécialisé (*crawler*) capable de récupérer sur le Web des documents relevant d'une thématique donnée, la plate-forme d'annotation qui enrichit les documents d'annotations sémantiques, un module d'analyse thématique des documents qui sert notamment pour les calculs de pertinence, un module d'indexation et une interface qui donne accès à des fonctions avancées de recherche et d'affinement de requêtes. La figure 1 présente l'architecture globale d'un moteur spécialisé construit à partir de ces modules.

Nous mettons ici l'accent sur l'annotation des documents. Il s'agit d'enrichir les documents source en anglais et en français, avec des annotations sémantiques. Dans l'architecture globale du moteur de recherche spécialisé, ces annotations sont ensuite prises en compte au niveau de l'indexation et rendues accessibles à l'utilisateur *via* l'interface. Celle-ci présente un index sémantique des documents retrouvés et des fonctionnalités d'affinement sémantique de requêtes. La figure 2 montre le type d'annotations produites : les termes et les entités nommées, qui sont considérées ici comme les unités sémantiques, sont identifiés, normalisés et typés sémantiquement. La production de ces annotations sémantiques présuppose par ailleurs des annotations morphosyntaxiques ou syntaxiques. C'est donc un traitement linguistique assez riche des documents qu'il faut effectuer.

L'intégration de techniques de TAL en recherche d'information a souvent donné des résultats décevants ou difficiles à interpréter : « *the impact of NLP on information*

3. L'ensemble de nos exemples de biologie ont été fournis par l'équipe INRA/MIG qui a développé et testé un moteur de recherche spécialisé en microbiologie intégrant la plate-forme Ogmios (voir (Buntine *et al.*, 2007) et section 5.2).
4. Projet européen IST / STREP n° 002068, voir `http://kearsage.hiit.fi/alvis`.

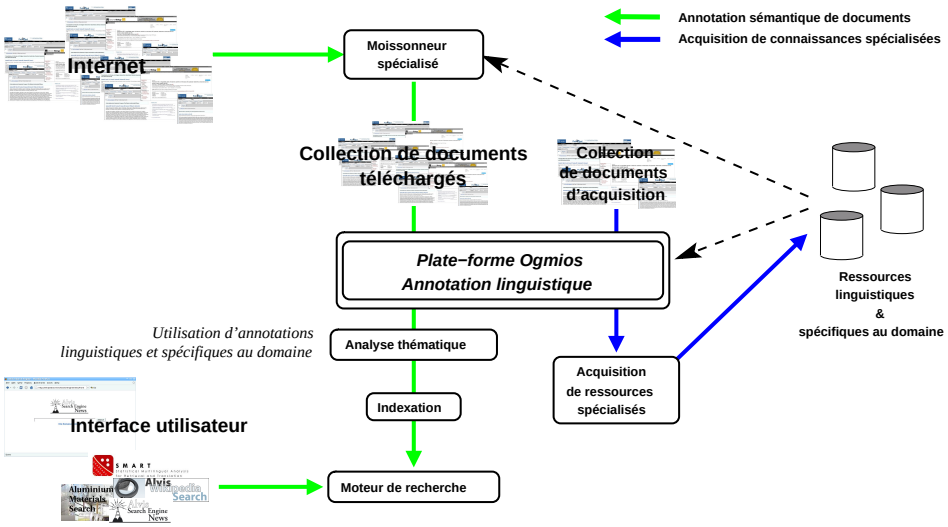

Figure 1. *Place de la plate-forme Ogmios dans l'architecture d'un moteur de recherche spécialisé*

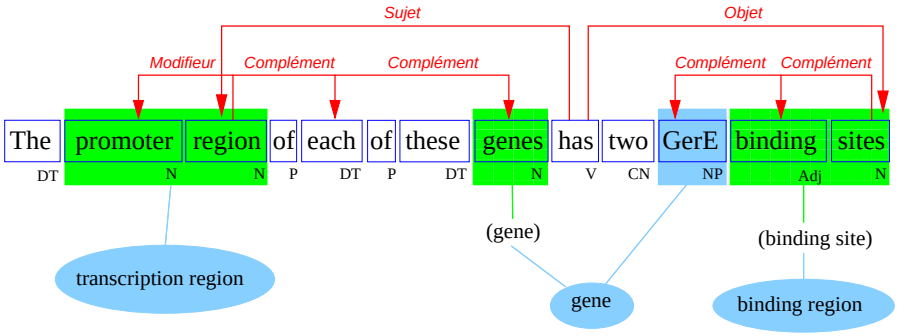

Figure 2. *Annotations produites par la plate-forme (les termes sont identifés par des rectangles verts, les formes canoniques sont entre parenthèses, les étiquettes sémantiques sont symbolisées dans les losanges bleus et les dépendances syntaxiques sont identifiées à l'aide de flèches rouges)*

retrieval tasks has largely been one of promise rather than substance » (Smeaton, 1997). La question se pose cependant différemment ici dans la mesure où la recherche d'information est ciblée sur un domaine, une collection de documents et un certain type de besoins. Nous faisons l'hypothèse que le domaine définit un champ pour lequel il est possible de représenter et de convoquer des connaissances sémantiques.

Le module d'annotation prend donc en entrée des documents issus du Web nettoyés et mis au format XML et leur associe des annotations de différents types. Les *annotations* sont des étiquettes associées à un segment textuel ou, plus rarement, à des couples de segments textuels dans le cas d'annotations relationnelles comme les dépendances syntaxiques, par exemple. Ces étiquettes s'interprètent en référence à des connaissances linguistiques ou du domaine. Le processus d'annotation repose ainsi sur des modules de traitement (des *outils*) qui font eux-même appel à des connaissances extérieures considérées comme des *ressources* par rapport aux modules de traitement.

2.2. *Pourquoi développer une plate-forme spécifique ?*

Dans ce contexte, il est important de disposer de plates-formes d'annotation pour concevoir et mettre au point des modules d'annotation adaptés aux moteurs de recherche spécialisés. Les chaînes de traitements à mettre en œuvre diffèrent en effet en fonction de la richesse de l'annotation recherchée, de la taille de la collection à traiter, du domaine à traiter et des besoins des utilisateurs, des outils et ressources disponibles.

Le TAL atteint aujourd'hui un nouveau stade de maturité qui permet l'émergence de telles plates-formes : ses techniques et ses résultats sont progressivement rendus accessibles à des non-spécialistes. On en veut pour preuve la généralisation de l'utilisation de WordNet ou la récente émergence de plates-formes de TAL, phénomène dont l'entreprise de GATE (Bontcheva *et al.*, 2004) a été précurseur et dont le présent numéro de la revue est le reflet. En dépit de ces progrès, il a fallu développer dans ALVIS une plate-forme spécifique pour répondre aux objectifs présentés ci-dessus.

La première raison était liée, en 2004, à des questions de performance. Des tests nous ont montré que GATE ne convenait pas, à l'époque, au traitement de gros corpus de documents : seuls de petits volumes de documents pouvaient être traités sans rencontrer des problèmes. GATE ayant été conçu au départ comme un environnement puissant de conception et de développement d'applications de TAL dans le cadre de l'extraction d'information, le passage à l'échelle n'était pas un objectif central. La *métaplate-forme* KIM (Popov *et al.*, 2004), qui s'appuie sur GATE, a été proposée pour satisfaire cette contrainte dans le cadre de projets d'annotations sémantiques massives SWAN[5] et SEKT[6]. Malheureusement, si les auteurs identifient le passage à l'échelle comme un paramètre critique, aucune mesure de performance n'est fournie concernant le temps de calcul ou le volume de documents traités. L'alternative de UIMA est intéressante pour le traitement de grandes collections. Cette plate-forme offre la possibilité de traiter les documents les uns après les autres ou sous forme d'une collection, un *Collection Processing Engine* (CPE) gérant la parallélisation et le contrôle des performances. Nos travaux se sont déroulés en parallèle du développement de UIMA et sur des principes assez proches concernant la parallélisation et

5. `http://deri.ie/projects/swan`
6. `http://sekt.semanticweb.org`

l'encapsulation des traitements. Les premiers résultats obtenus avec UIMA devraient permettre prochainement de mieux comparer les approches[7].

Ogmios se distingue par ailleurs des plates-formes comme LinguaStream (Widlöcher *et al.*, 2005), qui est conçu comme un outil de dépouillement de corpus et d'expérimentations. L'ambition est moindre : Ogmios ne vise pas à formaliser des traitements aussi complexes. La destination est également différente : la plate-forme Ogmios est conçue pour des concepteurs d'applications et non pour des linguistes ou à des fins d'analyse de corpus. L'accent est mis sur la robustesse des traitements au détriment de leur richesse et de leur complexité.

Nous nous sommes néanmoins appuyés sur l'existant à plusieurs égards :

– le problème de la représentation et de l'encodage des annotations linguistiques a été amplement étudié depuis le début des années 1990 et plusieurs formats ont été proposés (Grishman, 1997; Bird *et al.*, 1999). Des efforts sont faits pour unifier ces différents formats en vue d'améliorer l'interopérabilité des outils de TAL. Une proposition ISO (TC37SC4/TEI) est en cours de définition (Ide *et al.*, 2004). Nos objectifs sont moins ambitieux mais nous nous sommes inspirés de cette proposition, notamment en ce qui concerne l'utilisation de « tokens » comme référence pour l'ensemble des annotations ;

– il existe aujourd'hui des outils éprouvés et librement disponibles qui permettent de construire des plates-formes de TAL sans pour autant redévelopper tous les modules de traitement. C'est l'approche que nous avons retenue chaque fois que cela était possible (voir section 4.2) ;

– grâce aux campagnes d'évaluation probablement, on a aujourd'hui une définition plus stable et plus consensuelle des tâches de TAL. On progresse peu à peu dans le sens d'une décomposition standardisée des traitements (étiquetage des entités nommées, étiquetage morphosyntaxique, analyse syntaxique, etc.) qui facilite leur enchaînement et le remplacement d'un module par un autre ;

– un autre mouvement important des années 1990 est l'attention portée aux ressources, mouvement marqué par la création de centres de ressources comme le *Linguistic Data Consortium*[8] ou ELDA[9]. Des ressources de plus en plus nombreuses sont recensées même si elles ne sont pas toujours libres. Notre approche s'inscrit dans la même démarche consistant à séparer au maximum les modules de traitement et les ressources qu'ils exploitent. Nous verrons plus loin que la plate-forme Ogmios peut être utilisée aussi bien pour créer de nouvelles ressources que pour annoter des documents à l'aide de ressources existantes.

---

7. Voir l'atelier associé à la conférence LREC2008 : « *Towards Enhanced Interoperability for Large HLT Systems : UIMA for NLP* ».
8. `http://www.ldc.upenn.edu/`
9. `http://www.elda.org/`

3. Des verrous à lever

Les objectifs définis ci-dessus ont imposé de prendre en compte un certain nombre de contraintes dans le développement de la plate-forme. Nous les analysons ici plus en détail en expliquant comment elles ont été prises en compte.

3.1. *Contraintes opérationnelles*

Les premières contraintes sont opérationnelles : l'analyse des documents issus du Web doit être robuste et efficace, les données du Web étant généralement hétérogènes et volumineuses.

Les documents à traiter sont hétérogènes dans leur encodage, leur format, leur taille et leur qualité rédactionnelle, une diversité qui défie la robustesse de la plate-forme utilisée : même si la qualité du traitement s'en ressent, il faut pouvoir analyser sans blocage des flux de documents dont certains sont très longs, avec beaucoup de textes, et d'autres très courts ou avec surtout des tableaux, qui utilisent différents jeux de caractères et différentes langues parfois au sein du même document. Pour homogénéiser ce flux, le choix a été fait de nettoyer les documents en amont pour les mettre au format XML et de les encoder tous en UTF-8 quel que soit le jeu de caractères initial.

Ogmios étant conçu pour un moteur de recherche, il y a aussi des contraintes d'efficacité à prendre en compte. Les traitements de TAL sont généralement coûteux en temps de calcul même si des progrès ont été faits. Dans le cadre d'ALVIS, le module d'analyse des documents doit rester en phase avec le moissonneur qui récupère des documents. Pour répondre à cette exigence d'efficacité, nous n'avons pas cherché à optimiser les outils utilisés : nous cherchions à construire un prototype de recherche et nous avons misé, au contraire, sur la réutilisation des outils existants. En rechanche, nous avons mis en place une architecture client-serveur pour distribuer les traitements des flux entrants de documents sur autant de machines que nécessaire.

3.2. *Contraintes d'interopérabilité*

Rendre les outils de TAL interopérables a imposé un autre ensemble de contraintes qui concernent la modularité et la coopération des traitements ainsi que la normalisation de leurs formats d'entrée/sortie.

Permettre l'enchaînement et la substitution des outils suppose d'avoir décomposé le traitement global en modules élémentaires les plus standard possibles. Or, on constate que les outils disponibles assurent souvent plusieurs traitements élémentaires : TreeTagger (Schmid, 1997) découpe le texte en phrases avant d'associer des étiquettes morphosyntaxiques aux mots ; certains analyseurs syntaxiques gèrent eux-mêmes l'étiquetage morphosyntaxique et ont leur propre stratégie de reconnaissance des mots composés. Dans Ogmios, nous avons veillé à préserver la modularité des traitements, quitte, parfois, à désactiviter ou à contourner certaines fonctions des ou-

tils utilisés. Nous avons, par exemple, configuré TreeTagger pour qu'il exploite le découpage en phrases effectué par un module indépendant. De la même manière, nous avons proposé d'utiliser le Link Grammar Parser (Sleator *et al.*, 1993) uniquement pour l'analyse des dépendances syntaxiques à distance en confiant les tâches d'étiquetage morphosyntaxiques et d'analyse terminologique à des modules spécifiques.

Le fait de réutiliser des outils développés dans des contextes différents complique par ailleurs l'alignement de leurs résultats : tous les outils ne reposent pas sur la même conception du mot et de la phrase, par exemple. Là encore, la modularité permet un meilleur partage des tâches. En pratique, quand il n'était pas possible de décomposer suffisamment les outils utilisés, il a fallu aligner leurs résultats sur des annotations de référence : ces problèmes d'alignement concernent au premier chef des caractères d'espacement absents ou surnuméraires, la gestion différente des marques de ponctuation, le découpage en mots qui varie d'un outil à l'autre, etc. Pour effectuer ces alignements proprement, la plate-forme Ogmios repose sur une segmentation de référence. Le premier traitement effectué sur le texte d'entrée consiste à le découper en unités de base, des tokens, ce qui permet de définir des *offsets* (indices délimitant une séquence en nombre de caractères par rapport au début du document) pour garantir l'homogénéité des différentes annotations. Ce découpage repose sur un simple calcul sur les chaînes de caractères, il n'a aucune valeur linguistique. Il établit la référence par rapport à laquelle toutes les autres annotations sont exprimées directement ou indirectement.

Le problème de la coopération entre les outils est simplifié par ailleurs du fait que nous ne prenons pas en compte les annotations concurrentes, l'accent étant surtout mis sur la robustesse des traitements. En cas d'incohérence entre deux modules de traitement, les annotations du premier sont corrigées par le second.

Chaque outil ayant généralement ses formats propres, il est crucial de définir un format d'échange permettant d'interconnecter librement des outils. Nous avons donc défini un format homogène d'encodage des annotations qui repose sur le langage de balises XML (Nazarenko *et al.*, 2006a)[10]. Nous avons choisi de déporter les annotations hors du texte initial pour préserver l'intégrité du document. Dans l'architecture des moteurs d'ALVIS, le format déporté permet aussi d'exploiter indépendamment les annotations et le texte original. Les premières sont utilisées par le module d'indexation alors que le second doit être visualisable dans l'interface. Le langage XML étant cependant très verbeux, le volume de données à stocker est considérable, ce qui peut nécessiter une infrastructure matérielle adaptée. Le format que nous avons défini prend en compte différents niveaux d'unités textuelles. Le texte est d'abord découpé en tokens et toutes les autres unités (mots, syntagmes, unités sémantiques, phrases) sont construites sur ce premier niveau. À ces unités, peuvent être attachées des propriétés et des relations, morphologiques, syntaxiques et sémantiques. En pratique, intégrer un nouvel outil dans la plate-forme revient à définir un module qui l'encapsule (un *wrap-*

10. La DTD complète est définie dans (Nazarenko *et al.*, 2004) et (Taylor, 2007).

per) et qui assure la conformité de ses entrées/sorties avec le format d'annotation de la plate-forme en gérant la traduction vers et depuis le format propre de l'outil.

3.3. *Exigence de spécialisation*

Un autre contrainte forte est imposée par la nécessité d'adapter les traitements à des collections de documents qui peuvent elles-mêmes être spécialisées. On sait, en effet, que les performances des outils génériques de TAL se trouve dégradées quand ils sont utilisés sur des domaines et des sous-langages particuliers. Dans ALVIS, l'objectif étant de développer des briques logicielles pour des moteurs de recherche spécialisés, la plate-forme d'annotation doit être générique mais spécialisable. Deux voies de spécialisation sont possibles. On peut d'abord remplacer les outils par défaut par des outils spécialisés, à supposer qu'il existe effectivement des outils de TAL adaptés au domaine visé. C'est ce que nous avons fait ponctuellement pour certaines expériences : pour les textes de microbiologie, l'analyseur par défaut (Link Grammar Parser) a été remplacé par une version spécifiquement développée pour la biologie (BioLG[11] (Pyysalo *et al.*, 2006)). L'autre approche consiste à fournir aux outils des ressources adaptées au domaine à traiter (dictionnaires d'entités nommées, terminologies, thesaurus, ontologies). Cette approche a le mérite de la simplicité, les traitements et les ressources étant bien distingués, mais elle impose de développer en parallèle les ressources nécessaires. De ce point de vue, annoter les documents et acquérir les connaissances nécessaires à l'annotation sont deux tâches intimement liées.

La plate-forme Ogmios est conçue pour effecteur les deux types de traitement et c'est l'une de ses originalités. L'objectif principal est l'annotation des documents qui sont ensuite indexés dans l'architecture globale des moteurs d'ALVIS, mais dans la mesure où l'annotation dépend de ressources spécialisées, Ogmios prend également en charge l'acquisition de ces ressources. Dans ce processus d'acquisition, on constitue un corpus d'acquisition en choisissant un ensemble de documents représentatifs de la collection à analyser. Ce corpus est analysé et des ressources sont construites à partir des annotations résultantes soit simplement en extrayant le vocabulaire d'annotation, soit en généralisant les descriptions par des techniques d'apprentissage automatique, soit encore en combinant les approches précédentes avec une validation manuelle (Nazarenko *et al.*, 2006b). Le module d'étiquetage terminologique, qui repère les termes d'un corpus, exploite ainsi une ressource terminologique. Si on ne dispose pas de la ressource appropriée, on peut la construire en intégrant à la plate-forme Ogmios non pas un étiqueteur de termes mais un extracteur (Aubin *et al.*, 2006). L'extraction terminologique étant une opération d'acquisition plus complexe et plus coûteuse que l'étiquetage, il est utile de procéder à l'acquisition sur un petit corpus avant d'utiliser la terminologie résultante pour de l'annotation à grande échelle. L'acquisition de règles de reconnaissance d'entités nommées peut se faire de la même manière : des règles d'extraction sont apprises à partir d'un corpus déjà annoté par simple projection d'un premier échantillon d'entités nommées.

11. `http://www.it.utu.fi/biolg`

Il est important que les deux types de traitements soient effectués à partir de la même plate-forme et à l'aide des mêmes outils. Cela assure l'homogénéité des ressources produites avec les collections à annoter, et donc leur adéquation en termes de couverture de vocabulaire et de niveau d'expression des règles d'étiquetage. C'est le module final de la chaîne de traitements qui détermine si la sortie prend la forme d'un corpus annoté ou d'une ressource. L'autre différence tient souvent au volume de documents à analyser : les flux d'acquisitions étant moins importants que les flux d'annotations, des traitements plus lourds (comme l'analyse syntaxique) ou un processus de validation manuelle peuvent être envisagés.

3.4. *Contraintes de localisation*

Sans effectuer de traitement multilingue à proprement parler, La plate-fome Ogmios peut être paramétrée pour analyser des traitements dans différentes langues. Dans le cadre d'ALVIS, différentes chaînes de traitements ont été proposées pour l'anglais, pour le français, dans une moindre mesure, pour le slovène et, avec une décomposition assez différente, pour le chinois. Un traitement préparatoire du moissonneur détecte la langue du document. Cette annotation permet ensuite d'appliquer la bonne version d'Ogmios. À ce stade, ce « routage » s'opère au niveau du document, ce qui ne permet pas des traitements différenciés pour les documents hybrides qui contiennent des parties rédigées dans des langues différentes. Les exemples de cet article sont en anglais parce que c'est la version anglaise d'Ogmios qui a été la plus testée dans le cadre du projet ALVIS.

4. Présentation de la plate-forme Ogmios

La plate-forme Ogmios a donc été développée pour prendre en compte les contraintes ci-dessus.

4.1. *Architecture*

Dans la chaîne de traitements d'un moteur de recherche, la plate-forme d'annotation prend en entrée des documents Web téléchargés, nettoyés, codés en UTF-8 et convertis au format XML (Taylor, 2007). Un exemple de document en entrée de la plate-forme est présenté à la figure 11, en annexe.

Les documents sont ensuite traités par divers modules qui encapsulent des outils de TAL. La figure 3 présente l'architecture de la plate-forme. Les boîtes représentent les différents modules composant la chaîne de traitements linguistiques. Ces modules sont décrits dans la section 4.2. Les flèches vertes représentent le flux d'annotations des collections de documents issus du Web tandis que les flèches bleues représentent le flux d'acquisitions des ressources à partir de corpus, ces ressources étant ensuite utilisées dans la plate-forme (flèches en pointillé). La version actuellement disponible

intègre l'ensemble des traitements jusqu'à l'analyse syntaxique incluse. Les modules d'étiquetage sémantique et de résolution d'anaphore ne sont encore intégrés que de manière expérimentale.

Cette architecture n'est évidemment pas figée : si l'ordre des traitements est assez contraint, tous ne sont pas nécessaires pour toutes les applications. On peut procéder à l'analyse syntaxique sans analyse terminologique préalable. Nous intégrons la résolution des anaphores pour tester l'intégration de fonctionnalités avancées d'extraction d'informations relationnelles dans des moteurs de recherche spécialisés mais le module n'est que rarement exploité en recherche d'information.

4.2. *Chaîne de traitements*

La plate-forme Ogmios permet de créer des chaînes de traitements variés en utilisant des outils existants et en fonction des objectifs d'annotations. Nous présentons ici une version assez complète utilisant les modules intégrés par défaut pour l'anglais. Elle permet d'enchaîner les traitements décrits ci-dessous. Nous montrons sur des exemples les sorties (annotations) de ces différents traitements[12]. À noter que des outils similaires sont également intégrés pour le français (à l'exception de l'analyse syntaxique) et la conception de la plate-forme permet aisément la substitution d'un outil par un autre :

– le module assurant la *reconnaissance des entités nommées* identifie les séquences textuelles qui renvoient à une entité, leur associe un type sémantique et, le cas échéant, normalise cette séquence. Dans la suite des traitements, une entité nommée est considérée comme une seule unité et assimilée à un mot. Le module encapsule TagEN (Berroyer, 2004), qui repose essentiellement sur des dictionnaires et l'application de règles décrites sous forme de transducteurs. La figure 4 présente les entités nommées identifiées dans un extrait d'un résumé proposé par PubMed. Des types sémantiques biologiques (`gene` et `species`) leur sont associés ;

– le module de *segmentation en phrases et en mots* exploite un ensemble d'expressions régulières reprenant l'algorithme proposé dans (Grefenstette *et al.*, 1994). Ni les caractères de ponctuation, ni les caractères espace ne sont considérés comme des mots ;

– l'*étiquetage morphosyntaxique* repose sur la segmentation effectuée à l'étape précédente. C'est TreeTagger (Schmid, 1997) qui est utilisé comme étiqueteur par défaut. Nous avons aussi testé l'intégration de GeniaTagger (Tsuruoka *et al.*, 2005) qui est spécialisé pour la biologie mais le gain en qualité de l'étiquetage se fait au détriment des performances ;

– la *lemmatisation* associe un lemme à chaque mot du texte en s'appuyant sur l'analyse morphosyntaxique préalable. Si le mot ne peut pas être lemmatisé (nombres,

12. Pour des raisons de lisibilité, les exemples ne sont pas présentés dans le format réellement utilisé par la plate-forme.

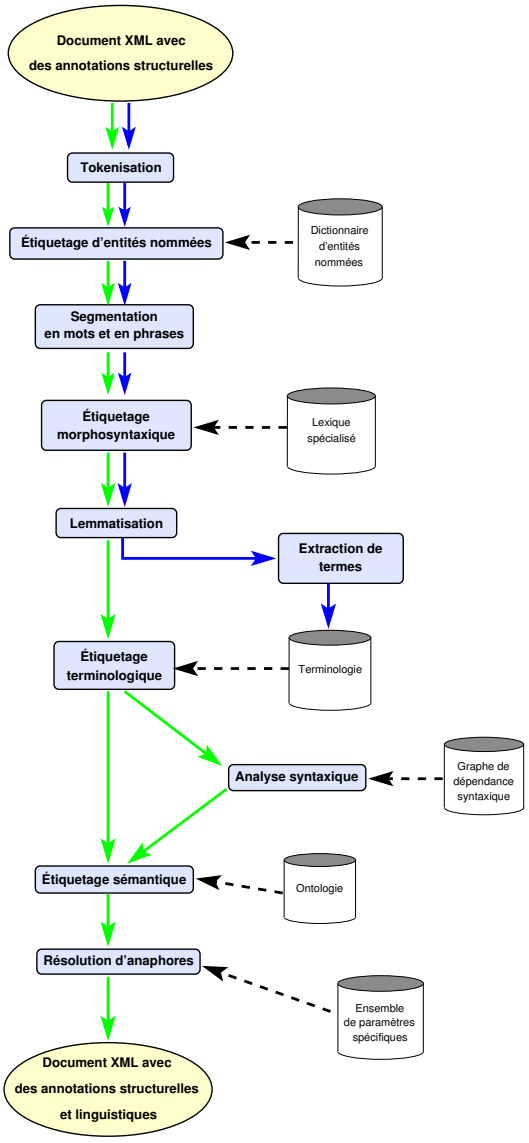

Figure 3. *Architecture de la plate-forme Ogmios*

During sporulation of <NE type=species>Bacillus subtilis</NE>, spore coat proteins encoded by <NE type=gene>cot</NE> genes are expressed in the mother cell and deposited on the forespore. Transcription of the <NE type=gene>cotB</NE>, <NE type=gene>cotC</NE>, and <NE type=gene>cotX</NE>genes by final <NE type=gene>sigma(K)</NE> RNA polymerase is activated by a small, DNA-binding protein called <NE type=gene>GerE</NE>. The promoter region of each of these genes has two <NE type=gene>GerE</NE> binding sites.

Figure 4. *Étiquetage des entités nommées*

mots étrangers, mots inconnus), aucune information n'est associée à la forme. L'intégration d'outils tels que TreeTagger ou GeniaTagger (réalisant à la fois l'étiquetage morphosyntaxique et la lemmatisation) nous a conduits à effectuer ces deux opérations en même temps, mais l'intégration d'un étiqueteur ne fournissant pas de lemmes, comme celui de Brill (Brill, 1995), nécessite de faire appel à un module spécifique pour la lemmatisation ;

– l'*étiquetage terminologique* repère les expressions du domaine qui ne sont pas des entités nommées, comme *gene expression* ou *spore coat cell* en biologie. L'analyse est réalisée en projetant les termes fournis en entrée sous la forme d'une ressource terminologique et en s'appuyant sur les résultats de l'analyse morphosyntaxique et de la lemmatisation. Dans la mesure où l'information est disponible dans la ressource, la structure syntaxique interne des termes est annotée. Dans la figure 5, les termes sont identifiés à l'aide de balises de type XML insérées dans le texte ;

During sporulation of Bacillus subtilis, <term>spore coat proteins</term> encoded by <term>cot genes</term> are expressed in the <term>mother cell</term> and deposited on the <term>forespore</term>. Transcription of the cotB, cotC, and cotX genes by final sigma(K) <term>RNA polymerase</term> is activated by a small, <term>DNA-binding protein</term> called GerE. The <term>promoter region</term> of each of these genes has two GerE <term>binding sites</term>.

Figure 5. *Étiquetage des termes*

– l'*analyse syntaxique* exploite les sorties de l'analyse morphosyntaxique. Nous avons choisi d'intégrer le Link Grammar Parser (Sleator *et al.*, 1993), qui repose sur des grammaires de dépendances, comme traitement par défaut. Pour le traitement des textes biomédicaux, c'est une version adaptée au domaine de la biologie BIOLG qui est utilisée (Pyysalo *et al.*, 2006) ;

– l'*étiquetage sémantique* projette un thesaurus sur le document pour associer des étiquettes sémantiques aux mots, aux entités nommées ou aux termes du texte. Se-

lon le type d'annotation disponible, ce module exploite ou non les dépendances syntaxiques à des fins de désambiguïsation. On peut acquérir le thesaurus de manière semi-automatique à partir de corpus (Cimiano, 2006; Aussenac-Gilles *et al.*, 2008) ou réutiliser une ressource existante, comme le métathesaurus UMLS (National Library of Medicine, 2003) pour les collections de documents en biologie et médecine. L'étiquetage sémantique des entités nommées et des termes identifiés dans l'extrait est présenté sous forme de balises XML, à la figure 6 ;

> During sporulation of <categoy semtag="species">Bacillus subtilis</category>, <categoy semtag="sporulation protein">spore coat proteins</category> encoded by <categoy semtag="gene">cot genes</category> are expressed in the <categoy semtag="cell component">mother cell</category> and deposited on the <categoy semtag="cell component">forespore</category>. Transcription of the <categoy semtag="gene">cotB</category>, <categoy semtag="gene">cotC</category>, and <categoy semtag="gene">cotX</category> genes by final sigma(K) <categoy semtag="enzyme">RNA polymerase</category> is activated by a small, <categoy semtag="binding region">DNA-binding protein</category> called <categoy semtag="gene">GerE</category>. The <categoy semtag="transcription region">promoter region</category> of each of these genes has two <categoy semtag="gene">GerE</category> <categoy semtag="binding region">binding sites</category>.

Figure 6. *Étiquetage sémantique*

– la *résolution des anaphores* exploite les résultats de tous les traitements précédents pour identifier les antécédents des pronoms. L'intégration du module de résolution de (Weissenbacher, 2008) est en phase de test. La plate-forme produit l'ensemble des indices (traits morphosyntaxiques de genre et/ou de nombre, rôles syntaxiques du pronom et de ses antécédents potentiels, catégories sémantiques de ces derniers, critères de saillance, etc.) qui sont utiles à la résolution d'anaphores. Les expériences de (Weissenbacher, 2008) montrent une qualité de résolution conforme à celle de l'état de l'art sur un corpus de biologie comportant 3 347 pronoms *it*, ce qui atteste de la relative qualité des annotations produites automatiquement par la plate-forme. La figure 7 présente l'identification d'une relation d'anophore à l'aide de balises XML.

> <anaphora id=1 type=antecedent>Another ORF</anaphora> apparently in the same transcription unit was found downstream from the amylase gene. <anaphora id=1 type=pronoun>It</anaphora> encoded a protein that was closely related to the maltose-binding protein of Escherichia coli.

Figure 7. *Identification des anaphores*

Les modules sont appelés les uns après les autres pour chaque document. Cette architecture séquentielle est assez traditionnelle mais certains points méritent d'être soulignés :

– la tokenisation constitue la première étape de la chaîne. Elle procède à une segmentation de base qui sert de référence pour les autres annotations[13]. Pour simplifier les traitements suivants, nous distinguons quatre types de tokens (alphabétiques, numériques, séparateurs ou symboliques) selon les séquences de caractères qui les composent. La figure 8 présente un exemple de tokenisation. Les barres obliques marquent les séparations entre les tokens (les caractères espace sont également des tokens) ;

/Transcription/ /of/ /the/ /cotB/,/ /cotC/,/ /and/ /cotX/ /genes/ /by/ /final/ /sigma/(/K/)/ /RNA/ /polymerase/ /is/ /activated/ /by/ /a/ /small/,/ /DNA/-/binding/ /protein/ /called/ /GerE/./

Figure 8. *Tokenisation*

– l'étiquetage des entités nommées est effectué très tôt dans la chaîne de traitements car l'identification des entités nommées désambiguïse certaines marques de ponctuation et prépare ainsi la segmentation en mots ou en phrases. On identifie ainsi les points d'abréviation (par ex. « *B. subtilis* » employé pour « *Bacillus subtilis* ») qui ne sont donc pas des marques de fin de phrases. De la même manière, les parenthèses figurant à l'intérieur d'un nom de gène comme *sigma(K)* ne sont pas considérées comme des caractères séparateurs. Si des informations morphosyntaxiques sont nécessaires pour la reconnaissance de certaines entités nommées, une deuxième étape d'étiquetage des entités nommées peut être effectuée après l'étiquetage morphosyntaxique ;

– l'étiquetage terminologique est utilisé tel quel mais peut également être considéré comme préalable à l'analyse syntaxique. Cette dernière étant coûteuse en temps de calcul, nous exploitons le fait que l'étiquetage terminologique revient à simplifier la phrase, réduisant d'autant l'ambiguïté et la complexité de l'analyse (Aubin *et al.*, 2005). On voit ici l'intérêt de la décomposition des traitements en tâches élémentaires. L'analyse syntaxique se focalise sur l'analyse des dépendances à distance et est ainsi déchargée de l'analyse plus locale des syntagmes nominaux complexes qui est assurée par l'étiqueteur terminologique, comme elle est traditionnellement déchargée des problèmes d'ambiguïté catégorielle qui sont pris en charge par l'étiquetage morphosyntaxique ;

– la même plate-forme peut être configurée soit pour annoter des documents, avec différents niveaux d'étiquetage, soit pour acquérir des ressources. Il suffit pour cela de remplacer le dernier module d'étiquetage de la chaîne par un module d'acquisition. Par exemple, pour l'acquisition de ressources terminologiques propres à un domaine,

13. Conformément aux recommandations du groupe TC37SC4/TEI, même si nous employons le terme d'*offset de caractère* plutôt que celui de *pointeur d'élément* pour désigner les frontières de chaque token.

le module d'étiquetage terminologique est remplacé par un module qui encapsule l'extracteur de termes YaTeA (Aubin *et al.*, 2006) (ce flux d'acquisitions est représenté par les flèches bleues de la figure 3).

4.3. *Implémentation*

La plate-forme est implémentée en Perl et est disponible sous forme de module CPAN (`http://search.cpan.org/~thhamon/Alvis-NLPPlatform/`). Elle analyse la collection document par document. Les sorties (annotations) sont stockées en mémoire jusqu'à la fin du traitement du document en cours. La structure de données interne utilisée facilite l'accès par référence aux unités textuelles et aux propriétés associées. Quand le traitement d'un document est terminé, toutes ses annotations sont enregistrées dans le format XML de la plate-forme. Un extrait de document annoté est présenté à la figure 12, en annexe. Le format utilisé est très verbeux (sur les expériences que nous avons faites, l'annotation augmente la taille de la collection d'un facteur 19,63). Il pourrait facilement être compressé[14], ce que n'avons pas cherché à faire pour le moment.

La plate-forme peut être exploitée soit de manière autonome, soit en modèle client/serveur. Ce dernier mode d'utilisation est particulièrement adapté à la répartition du moteur de recherche sur plusieurs machines. Il est ainsi possible d'adapter la puissance de calcul nécessaire à l'analyse de gros volumes de données sans que l'annotation linguistique constitue un point bloquant. En mode client/serveur, chaque client exécute une instance de la plate-forme. Les documents à traiter sont récupérés les uns après les autres par le client qui en fait la demande auprès du serveur. Le serveur distribue les documents à annoter et reçoit les documents annotés. Le dialogue qui en résulte entre le serveur et le client est implémenté sous la forme d'un protocole au dessus de TCP/IP. De plus, le serveur conserve les documents à annoter tant qu'il n'a pas réceptionné le résultat de l'annotation par le client. Lorsque la plate-forme Ogmios est utilisée dans le moteur de recherche Alvis, le serveur joue également le rôle d'intermédiaire entre le moissonneur spécialisé et le moteur d'indexation. L'utilisation de la plate-forme dans le mode autonome donne une vue globale de la collection des documents traités. Cette vue n'est pas disponible en mode client/serveur actuel. Il faudrait pour cela faire évoluer l'architecture, notamment pour différencier les clients et adapter le protocole de communication (nous revenons sur ce point en section 6.4).

Une alternative à l'exécution d'une instance de la plate-forme sur chaque client serait d'attribuer un traitement linguistique spécifique (segmentation en mots, étiquetage morphosyntaxique, etc.) à chaque client. Ce cas de figure augmenterait considérablement le volume de messages échangés entre les clients, éventuellement une utilisation excessive des processeurs pour l'analyse et la reconstruction des structures de don-

14. La compression à l'aide de l'outil `gzip` permet, par exemple, de réduire le ratio à 7,49 mais au détriment des temps de calcul et de la facilité d'accès aux annotations.

nées[15] et pourrait entraîner le blocage temporaire de la plate-forme sur des documents volumineux ou nécessitant beaucoup de temps de calcul. La plate-forme serait alors sous-utilisée, les clients en aval de la chaîne de traitements ne pouvant rien traiter en attendant la fin des traitements précédents. La solution que nous avons choisie limite au contraire les échanges et permet de contenir le blocage dû au traitement d'un document de grande taille à un seul client, les autres clients pouvant continuer à traiter d'autres documents. La robustesse des traitements est assurée en cas d'une interruption de ceux-ci due à l'arrêt d'une machine ou d'un processus à la suite d'un événement système. Le mode de distribution choisi permet enfin d'envisager un équilibrage des charges en fonction des capacités de traitements des machines et de la taille des documents.

4.4. *Mise en œuvre*

L'installation et l'utilisation de la plate-forme ont été testées sur des machines Linux Debian par plusieurs partenaires du projet ALVIS. Le dépôt de modules sur CPAN permet également d'accéder à des tests d'installation sur différentes versions de systèmes d'exploitation. Ainsi, l'installation des modules CPAN de la plate-forme a été testée avec succès sur Linux, Solaris, FreeBSD et Windows[16], l'utilisation du langage Perl assurant la portabilité de la plate-forme sur différents environnements.

L'exploitation de la plate-forme en tant que telle dépend des outils utilisés et de leur disponibilité sous le système considéré. Ces tests d'installation n'incluent donc pas l'installation des outils intégrés dans la plate-forme, qui doit être faite indépendamment. Actuellement la récupération et l'installation des outils sont documentées, mais ne sont pas encore automatisées (nous l'envisageons).

L'intégration d'un nouvel outil demande généralement le développement d'un nouveau *wrapper*, dont le coût varie suivant la complexité des formats d'entrées et surtout de sorties. Elle nécessite une connaissance approfondie de l'outil, certains comportements pouvant ne pas être correctement documentés. Les données en entrée des outils sont produites à partir des index internes de la plate-forme. L'exploitation des sorties et leur intégration comme annotations sont généralement réalisées en deux étapes. Les données en sortie sont, d'abord, réalignées avec celles de l'entrée pour obtenir les références aux unités linguistiques sur lesquelles vont s'appuyer les nouvelles annotations. Les informations produites sont ensuite intégrées dans les structures de données. L'ensemble des *wrappers* développés est livré et installé avec la plate-forme.

---

15. L'utilisation du protocole `SOAP` permettrait de s'affranchir de ces contraintes dans une certaine mesure, mais le transfert de messages volumineux reste problématique.
16. Voir `http://cpantesters.perl.org/show/Alvis-NLPPlatform.html`.

5. Expériences

Cette section décrit plusieurs expériences d'utilisation de la plate-forme Ogmios. Les premières ont permis de tester ses performances. Les suivantes montrent ce qu'apportent les annotations sémantiques calculées, à des moteurs de recherche spécialisés.

5.1. *Analyses de performance*

La plate-forme que nous avons développée vise à analyser des textes provenant du Web. Elle est conçue pour être intégrée à des moteurs spécialisés dans des domaines techniques. Même si les documents ne sont pas analysés au moment de la requête mais avant d'être indexés, les performances doivent être compatibles avec le rythme de récupération des documents sur le Web. On vise ainsi l'analyse de plusieurs Go de données par jour, un niveau de performance qui implique une architecture distribuée permettant d'ajouter de nouvelles machines en fonction de la charge.

Nous avons annoté deux collections de documents issus du Web. La première collection regroupe 55 329 documents biomédicaux (désormais BIO). La plupart des documents XML ont une taille comprise entre 1 ko et 100 ko. La taille du plus grand document est 5,7 Mo. L'hétérogénité de cette collection est illustrée par la figure 9 qui présente la distribution de la taille des documents. La seconde collection comporte 48 422 dépêches relatives aux moteurs de recherche (désormais SEN). La taille des documents varie entre 1 et 150 ko.

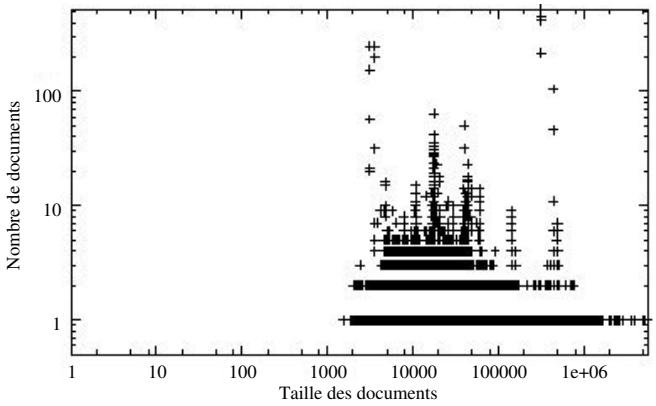

Figure 9. *Distribution de la taille des documents de la collection* BIO

Nous avons réalisé l'ensemble des traitements jusqu'à l'étiquetage terminologique. Pour l'annotation de la collection BIO, nous avons exploité une liste de 375 000 termes issus du MeSH (MeSH, 1998) et de Gene Ontology (Consortium, 2001). Pour la collection SEN, la terminologie comportait 17 341 termes extraits au-

tomatiquement. Nous avons utilisé une liste d'environ 400 000 entités nommées (noms d'espèces et de gènes sur le corpus BIO, noms de personnes, de logiciels et de sociétés sur le corpus SEN).

L'annotation des documents a été distribuée sur vingt ordinateurs. La plupart sont des ordinateurs classiques (de type PC) avec 1 Go de mémoire vive (RAM) et un processeur cadencé à 2,9 ou 3,1 GHz. Nous avons également utilisé un ordinateur avec 8 Go de RAM et deux processeurs Xeon cadencés à 2,8 GHz (processeur Xeon dual core). Le système d'exploitation est Linux (Debian ou Mandrake). Le serveur et trois clients étaient hébergés sur la machine biprocesseur Xeon. Chaque ordinateur personnel abritait un seul client réalisant l'ensemble de la chaîne de traitements.

Les performances obtenues donnent une bonne idée des performances globales de la plate-forme même si une évaluation complète aurait demandé des séries de tests plus importantes. L'annotation de la collection BIO a été effectuée en 35 heures[17]. Le corpus est composé de 106 millions de mots et 4,72 millions de phrases. 147 documents ne contenaient aucun mot, ils n'ont donc pas été analysés au-delà de l'étape de tokenisation. Un des clients a analysé un document composé de 414 995 mots. En moyenne, un document est analysé en 35 secondes et la génération du fichier XML prend 2 secondes supplémentaires. Les étapes les plus coûteuses en temps de traitement sont celles qui demandent le plus de ressources : la reconnaissance des termes (56 % du temps de traitement global) et la reconnaissance des entités nommées (16 % du total).

Ces collections de documents ont été traitées sans rencontrer de problème. Les performances obtenues montrent que la plate-forme développée est robuste et qu'elle peut traiter des grandes masses de textes dans des temps raisonnables. Ces performances globales pourraient être encore améliorées par une optimisation du code de la plate-forme, un équilibrage de la charge afin de proposer aux clients des documents dont la taille est adaptée à leur capacité de traitement et le remplacement de certains modules de traitement par des outils plus efficaces.

5.2. *Apport de l'annotation dans les moteurs spécialisés*

Dans le cadre du projet ALVIS, la plate-forme Ogmios a été utilisée par d'autres partenaires pour construire différents moteurs de recherche[18], mais l'exemple le plus abouti est celui qui a été développé par l'équipe INRA/MIG pour la microbiologie. La figure 10 donne un exemple des résultats fournis par ce moteur BioAlvis[19]. Sans faire une évaluation réelle de l'apport de l'analyse préalable des documents, nous montrons sur quelques exemples qu'elle permet de répondre à des besoins d'informations spé-

17. Le temps d'exécution de chaque module a été enregistré à l'aide du module Perl `http://search.cpan.org/~jhi/Time-HiRes/`. Les temps d'analyse sont enregistrés dans le fichier XML produit en sortie.
18. Voir par exemple `http://wikipedia.hiit.fi/searchenginenews/front`.
19. http ://genome.jouy.inra.fr/alvis/front

cialisés. Nous nous appuyons pour ce faire sur les tests réalisés par les biologistes de l'INRA (Buntine *et al.*, 2007).

Figure 10. *Résultat obtenu par le moteur BioAlvis pour la requête «* Stress facter Escherichia coli *»*

Les différentes catégories d'entités nommées repérées dans les documents retournés par le moteur figurent sous la forme de différents index en partie gauche de l'interface. Ils permettent à l'utilisateur de focaliser sa recherche sur une entité particulière. À partir de la requête *stress factor*, on peut rapidement mettre l'accent sur les facteurs de stress chez un bacille particulier nommé *Escherichia Coli*. Contrairement à d'autres moteurs de recherche, cette indexation des documents retrouvés n'est pas faite à la volée, elle exploite les annotations préalablement associées aux documents, ce qui laisse le temps de faire une analyse sémantique de meilleure qualité – même si elle reste assez fruste.

L'annotation sémantique permet aussi de traiter les problèmes d'ambiguïtés, qu'on trouve dans tous les domaines, aussi spécialisés soient-ils. Comme le nom de gène *red* est ambigu avec un nom de couleur, le biologiste obtient évidemment une meilleure précision de résultat en cherchant *red* étiqueté en tant que nom de gène plutôt que comme un simple mot (49 documents retournés contre 147). L'interface du moteur permet de faire cela facilement.

Le traitement linguistique des documents permet également de prendre en compte les problèmes de variations et de retrouver les documents pertinents pour la requête de l'utilisateur même si les mots exacts de sa requête ne figurent pas dans le document. La

normalisation des termes effectuée lors de l'étiquetage terminologique permet ainsi de rapprocher des formes comme *gene expression* et *expression of gene(s)* ou *synthesis* et *biosynthesis*.

L'utilisation d'un thesaurus sémantique (en haut à gauche sur l'interface utilisateur) permet d'interroger sur les concepts plutôt que sur les mots. L'utilisation de termes plus spécifiques que *Stress factor* tels que *cold shock*, *high temperature*, *low temperature* ou *amino-acid starvation* permet d'augmenter le rappel du moteur (2 941 documents retournés par BioAlvis contre 870 par PubMed qui indexe pourtant une collection de taille supérieure). Seule une ressource sémantique spécialisée peut permettre de faire de tels calculs. Dans le cas de BioAlvis, un thesaurus sémantique organisant les termes de la collection en hiérarchie a été construit. Il est utilisé pour enrichir les requêtes. Ce thesaurus a été construit semi-automatiquement à partir de la terminologie extraite d'un corpus d'acquisition en s'appuyant sur la distribution syntaxique des mots. C'est un des cas de figure dans lequel l'analyse syntaxique du corpus d'acquisition est exploitée.

6. Nouveaux défis à relever

L'expérience du développement et du test d'une plate-forme d'annotation comme Ogmios montre que de nouveaux défis restent à relever. Nous en pointons quelques-uns ici.

6.1. *Évaluer la plate-forme*

Les expériences rapportées ci-dessus ne constituent pas une évaluation de la plate-forme. Celle-ci reste à faire. Dans la mesure où l'objectif était de rendre interopérables des outils existants plutôt que de développer des outils spécifiques, il faut évaluer la plate-forme indépendamment des performances propres des outils auxquels elle fait appel. À notre connaissance, il existe peu d'expériences visant à évaluer ainsi la qualité d'une plate-forme et de l'enchaînement des outils indépendamment de la qualité des outils particuliers utilisés. Les tests de régression de GATE constituent un pas dans cette direction, mais il s'agit d'un outil de diagnostic plus que d'analyse. Plusieurs approches doivent cependant être envisagées.

Il faut d'abord comparer les performances des outils pris individuellement et intégrés dans la plate-forme en prenant en compte, dans ce dernier cas, différentes configurations de modules. Cela revient en pratique à comparer la qualité des annotations produites dans les deux cas. Si l'étiquetage morphosyntaxique produit par Ogmios est de meilleure qualité que les résultats de l'étiqueteur pris isolément, cela montre sans doute l'intérêt du couplage de la reconnaissance des entités nommées et de l'étiquetage morphosyntaxique.

Une deuxième approche, complémentaire, consiste, pour certains types d'erreurs, à analyser le comportement de la plate-forme pour déterminer si elles font diverger

le système global ou si la plate-forme montre au contraire une certaine résistance au bruit. Dans le meilleur des cas, on devrait observer des processus de récupération sur erreurs. Étant donné, par exemple, une phrase où certains termes ont été identifiés par l'étiqueteur terminologique, s'il s'avère impossible de faire l'analyse syntaxique de cette phrase, c'est peut-être que l'analyse terminologique préalable est erronée et qu'il faut décomposer certains termes.

6.2. *Se doter d'outils d'expérimentation*

Même si la plate-forme Ogmios n'est pas conçue pour l'exploration de corpus et la mise au point de traitements linguistiques complexes, il est important de se doter d'outils d'expérimentation pour déterminer quelle configuration retenir pour analyser une collection donnée. Il s'agit bien évidemment de proposer une interface de configuration qui permette aisément de fixer les ressources à exploiter, de sélectionner et d'ordonner les traitements. Mais il faut plus fondamentalement se doter d'outils de mesure et d'analyse pour apprécier la qualité d'annotation obtenue avec une certaine configuration d'outils et de ressources, ainsi que l'impact de tout changement de configuration. Pour éviter de refaire systématiquement des expériences à grande échelle, il est possible dans Ogmios de sauvegarder, dans un fichier XML, les résultats intermédiaires et de relancer l'annotation à partir de l'étape correspondante. Il faudrait surtout élaborer des mesures et des procédures permettant de prédire la qualité d'annotation attendue pour une configuration de la plate-forme et un corpus donnés, avant de lancer l'expérience à grande échelle.

C'est un chantier important à ouvrir non seulement pour Ogmios mais plus largement pour la crédibilité scientifique et l'essor du TAL dans son ensemble, car on sait mal définir aujourd'hui le domaine d'application des outils de TAL.

6.3. *Intégrer les ressources*

Un autre défi consiste à intégrer les ressources utilisées par les différents modules d'annotation. Pour l'instant la plate-forme fait appel à des ressources disjointes (un dictionnaire d'entités nommées, une ou plusieurs terminologies, un thesaurus) et rien ne garantit qu'elles soient cohérentes entre elles. Il faudrait penser le processus d'acquisition de manière plus globale pour s'assurer que les premières ressources acquises (dictionnaires d'entités ou terminologies) soient fournies en entrée du processus d'acquisition du thesaurus ou, ce qui peut être plus facile, s'assurer que le thesaurus est acquis à partir d'un corpus dans lequel les entités nommées et les termes sont annotés conformément aux ressources utilisées.

Un second défi, lié au précédent, consiste à définir un modèle cohérent de ressources qui intègre ou articule l'ensemble des connaissances exploitées dans le processus d'acquisition. On doit ainsi obtenir une ressource termino-ontologique associant un dictionnaire d'entités nommées, une terminologie et une ontologie du domaine

pour faire le lien entre les mots des textes à annoter et les concepts de l'ontologie ou leurs instances.

6.4. *Articuler les traitements locaux et globaux*

Distribuer le traitement des documents sur des machines différentes fait perdre de vue la notion même de collection de documents. Si certains traitements peuvent être effectués localement, *i.e.* à l'aune d'un document, d'autres nécessitent au contraire l'éclairage global de la collection de documents. C'est le cas, par exemple, des techniques de désambiguïsation endogène pour le calcul des rattachements syntaxiques (Bourigault, 1993) et des traitements qui exploitent des distributions de probabilités des unités du corpus.

Combiner les traitements locaux et les traitements globaux n'est pas trivial lorsque l'analyse des documents est répartie sur plusieurs clients. Plusieurs approches sont envisageables.

La première est exogène. Elle consiste à exploiter des connaissances extérieures. Celles-ci peuvent être issues des ressources existantes ou calculées sur un corpus d'acquisition.

Le calcul des connaissances peut être réalisé de manière endogène sur la collection en deux passes d'annotation ou bien en réalisant une annotation incrémentale. Dans le premier cas, chaque document pris individuellement est analysé. Les résultats obtenus sont capitalisés pour acquérir une vue globale de la collection qui est ensuite utilisée lors d'une deuxième passe d'annotation. L'annotation incrémentale consiste, là aussi, à capitaliser les éléments obtenus à partir de l'analyse des documents individuels, mais l'annotation se fait en une seule passe. Sans attendre d'avoir analysé toute la collection, on injecte sur un document $n + 1$ les connaissances obtenues à partir de la sous-collection composée des n documents précédemment analysés. Dans cette approche, tous les documents ne sont pas analysés de la même manière puisque la vue globale sur la collection évolue. Il faudrait vérifier que cette vue se stabilise néanmoins lorsque la taille de la collection augmente.

Dans le modèle client/serveur, les deux dernières approches supposent que le client chargé de l'analyse d'un document renvoie au serveur les informations qui en sont tirées pour permettre à ce dernier de construire incrémentalement une vue globale de la collection. C'est en quelque sorte la première approche qui est actuellement mise en œuvre dans Ogmios. Il faudrait tester les deux suivantes pour en mesurer le coût et l'intérêt.

6.5. *Définir une sémantique de l'annotation*

La plate-forme Ogmios implémente une vision très fruste de la sémantique, puisque le texte est décrit comme un ensemble d'atomes sémantiques (les termes et

les entités nommées) le plus souvent disjoints[20], mais celle-ci n'est pas clairement définie :

– dans l'hypothèse où toutes les ressources sont intégrées dans une unique ressource termino-ontologique cohérente (voir section 6.3), le processus d'annotation est guidée par l'ontologie du domaine. Cela signifie que les termes sont interprétés comme des étiquettes des concepts sollicités par le texte du document, ces concepts étant eux-mêmes structurés en ontologie. Les entités nommées quant à elles correspondent à des instances de concepts. En pratique, ces différents niveaux d'annotation sont confondus ;

– on ne distingue pas non plus les connaissances du domaine, qui sont utilisées sous la forme d'étiquettes sémantiques (quoi annoter ?), et les connaissances liées à la collection elle-même qui devraient permettent de contrôler le processus d'annotation (comment annoter ?).

Il conviendrait de formaliser davantage la sémantique construite par l'ensemble des annotations faites sur une collection de documents.

7. Conclusion

Cet article tire le bilan de l'expérience du développement de la plate-forme d'annotation de documents, Ogmios. Nous avons souligné les contraintes qui étaient les nôtres dans la perspective du développement de moteurs de recherche spécialisés : contraintes de performance et d'interopérabilité des outils du TAL, nécessité de spécialiser les traitements à façon pour analyser de collections centrées sur des domaines particuliers. Les solutions proposées consistent à distribuer les traitements sur plusieurs machines pour gagner en temps de traitement, à encapsuler les outils de TAL en travaillant sur un format d'annotation unique, les formats propriétaires des outils étant traduits dans un format d'échange et, enfin, à coupler le processus d'annotation avec des processus d'acquisition, les ressources ainsi construites étant ensuite utilisées dans l'annotation.

Le bilan qui ressort de ce travail de développement et de l'analyse des performances obtenues pour la plate-forme est contrasté. D'un côté, il est aujourd'hui envisageable de lancer des processus d'annotation sémantique à relativement grande échelle même sur des collections de documents hétérogènes comme celles qui sont construites à partir du Web. D'un autre côté, force est de constater que l'ingénierie du TAL reste délicate en dépit des progrès effectués. Les premières difficultés sont liées à la manière dont les outils gèrent les flux de caractères, ce qui se traduit par des problèmes d'alignement dans les résultats. D'autres difficultés tiennent à l'hétérogénéité des formats et des jeux d'étiquettes utilisés, lesquels doivent être unifiés, au dépens parfois de leur richesse. D'autres difficultés encore proviennent de l'insuffi-

20. À terme, on devrait ajouter dans la chaîne de traitements un module d'étiquetage des relations sémantiques à base de règles contextuelles, mais aujourd'hui il n'est pas encore intégré.

sante standardisation des tâches de TAL qui oblige à décomposer les outils avant de les recombiner entre eux. Ce travail d'ingénierie est néanmoins très utile pour la recherche parce qu'il permet de mettre en place rapidement des expérimentations avec des traitements spécialisés sur un domaine particulier et de tester ainsi des hypothèses de travail. Nous avons mentionné le travail effectué sur la résolution des anaphores (Weissenbacher, 2008). D'autres expériences sont en cours pour l'acquisition de ressources pour l'indexation contrôlée (Hamon *et al.*, 2008).

Cette expérience du développement d'Ogmios montre également que des défis importants doivent encore être relevés. Le premier concerne l'évaluation de ce type de plate-forme au-delà des mesures de performance et des tests d'utilisabilité. Développer la plate-forme pour en faire un véritable instrument d'expérimentation et de mise au point des traitements est un enjeu important pour l'ingénierie du TAL. L'analyse des processus sémantiques sous-jacents à l'annotation des documents constitue un autre défi, avec l'intégration des différentes ressources utilisées et l'articulation des traitements effectués localement au niveau du document et globalement au niveau de la collection.

Remerciements

Nous remercions les partenaires du projet ALVIS, notamment Claire Nédellec et Robert Bossy (INRA/MIG) qui ont développé le moteur de recherche BioAlvis et contribué à toutes les expériences sur le domaine de la biologie. De nombreuses discussions avec l'équipe INRA/MIG ont par ailleurs préparé le travail sur Ogmios. Nous remercions également Julien Derivière qui a participé au développement de la plateforme Ogmios quand il était ingénieur au LIPN.

8. Bibliographie

Aubin S., Hamon T., « Improving Term Extraction with Terminological Resources », *in* T. Salakoski, F. Ginter, S. Pyysalo, T. Pahikkala (eds), *Advances in Natural Language Processing (5th International Conference on NLP, FinTAL 2006)*, n° 4139 in *LNAI*, Springer, p. 380-387, August, 2006.

Aubin S., Nazarenko A., Nédellec C., « Adapting a General Parser to a Sublanguage », *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'05)*, Borovets, Bulgaria, p. 89-93, 2005.

Aussenac-Gilles N., Despres S., Szulman S., « The Terminae Method and Platform for Ontology Engineering from Texts », *in* P. Buitelaar, P. Cimiano (eds), *Bridging the Gap between Text and Knowledge : Selected Contributions to Ontology learning from Text*, IOS Press, 2008.

Berroyer J.-F., *« TagEN, un analyseur d"entités nommées : conception, développement et évaluation »*, Mémoire de D.E.A. d'intelligence artificielle, Université Paris-Nord, 2004.

Bird S., Liberman M., « Annotation graphs as a framework for multidimensional linguistic data analysis », *in* N. A. for Computational Linguistics (ed.), *Proceedings of the Workshop "Towards Standards and Tools for Discourse Tagging"*, Somerset, p. 1-10, 1999.

Bontcheva K., Tablan V., Maynard D., Cunningham H., « Evolving GATE to meet new challenges in language engineering », *Natural Language Engineering*, vol. 10, n° 3-4, p. 349-374, Sept-Dec, 2004.

Bourigault D., « An Endogenous Corpus-Based method for Structural Noun Phrase Disambiguation », *6th European Chapter of the Association for Computational Linguistics*, 1993.

Brill E., « Transformation-Based Error-Driven Learning and Natural Language Professing : A Case Study in Part-of-Speech Tagging », *Computational Linguistics*, vol. 21, n° 4, p. 543-565, december, 1995.

Buntine W. et al., Report on Tests, ALVIS Deliverable n° D8.3, ALVIS, March, 2007.

Cimiano P., *Ontology Learning and Population from Text*, Springer, 2006.

Consortium T. G. O., « Creating the Gene Ontology Resource : Design and Implementation », *Genome Res.*, vol. 11, n° 8, p. 1425-1433, 2001.

Grefenstette G., Tapanainen P., « What is a word, what is a sentence ? problems of tokenization », *The 3rd International Conference on Computational Lexicography*, Budapest, p. 79-87, 1994.

Grishman R., « Information Extraction : Techniques and Challenges », *in* M. T. Pazienza (ed.), *Information Extraction : a Multidisciplinary Approach to an Emerging Information Technology*, Springer, Berlin, p. 10-27, 1997.

Hamon T., Grabar N., « Acquisition of elementary synonym relations from biological structured terminology », *Proceedings of CICLing2008 - 9th International Conference on Intelligent Text Processing and Computational Linguistics*, Haifa, Israel, p. 40-51, 2008.

Ide N., Romary L., de la Clergerie E., « International standard for a linguistic annotation framework », *Natural Language Engineering*, vol. 10 (3/4), p. 211-225, 2004.

MeSH, « Medical Subject Headings », , Library of Medicine, Bethesda, Maryland, WWW page `http ://www.nlm.nih.gov/mesh/meshhome.html`,, 1998.

National Library of Medicine (ed.), *UMLS Knowledge Source*, 13^{th} edn, 2003.

Nazarenko A., Alphonse E., Aubin S., Deriviere K., Hamon T., Mladenic D., Nédellec C., Poibeau T., Weissenbacher D., Zhou Q., Report on augmented document representations, Deliverable n° D5.1, ALVIS, 2004.

Nazarenko A., Alphonse E., Derivière J., Hamon T., Vauvert G., Weissenbacher D., « The ALVIS Format for Linguistically Annotated Documents », *Proceedings of LREC 2006*, 2006a.

Nazarenko A., Nédellec C., Alphonse E., Aubin S., Hamon T., Manine A.-P., « Semantic Annotation in the Alvis Project », *Proceeding of IIA-2006 : International Workshop on Intelligent Information Access*, Helsinki, Finland, July, 2006b.

Popov B., Kiryakov A., Ognyanoff D., Manov D., Kirilov A., « KIM – a semantic platform for information extraction and retrieval », *Natural Language Engineering*, vol. 10, n° 3-4, p. 375-392, Sept-Dec, 2004.

Pyysalo S., Salakoski T., Aubin S., Nazarenko A., « Lexical adaptation of link grammar to the biomedical sublanguage : a comparative evaluation of three approaches », *BMC Bioinformatics*, November, 2006.

Schmid H., « Probabilistic Part-of-Speech Tagging Using Decision Trees », *in* D. Jones, H. Somers (eds), *New Methods in Language Processing Studies in Computational Linguistics*, 1997.

Sleator D. D., Temperley D., « Parsing English with a link grammar », *Third International Workshop on Parsing Technologies*, 1993.

Smeaton A. F., « Using NLP or NLP resources for information retrieval tasks », *in* T. Strzalkowski (ed.), *Natural language information retrieval*, Kluwer Academic Publishers, Dordrecht, NL, p. 99-111, 1997.

Taylor M., Report on metadata frameworks, including concrete representations, for network nodes and semantic document analyses, ALVIS Deliverable n° D3.1, ALVIS, 2007.

Tsuruoka Y., Tateishi Y., Kim J.-D., Ohta T., McNaught J., Ananiadou S., Tsujii J., « Developing a Robust Part-of-Speech Tagger for Biomedical Text », *Proceedings of Advances in Informatics - 10th Panhellenic Conference on Informatics*, LNCS 3746, p. 382-392, 2005.

Weissenbacher D., Influence des annotations imparfaites sur les systèmes de Traitement Automatique des Langues, un cadre applicatif : la résolution de l'anaphore pronominale, Thèse de doctorat en informatique, Université Paris 13, Novembre, 2008.

Widlöcher A., Bilhaut F., « La plate-forme LinguaStream : un outil d'exploration linguistique sur corpus », *Actes de la conférence TALN 2005*, Dourdan, France, p. 517-522, juin, 2005.

9. Annexe

```
<documentCollection>
<documentRecord id="A79ACA58DEB7E6114747710B9A85059F">
  <acquisition>
      <acquisitionData>
        <modifiedDate>2004-11-21 15:59:14</modifiedDate>
        <urls>
          <url>http://www.ncbi.nlm.nih.gov/pubmed/10788508?dopt=MEDLINE</url>
        </urls>
      </acquisitionData>
      <canonicalDocument>
        <section>
          <section title="Combined action of two transcription factors regulates
               genes encoding spore coat proteins of Bacillus subtilis.">
            <section>Combined action of two transcription factors regulates genes
                 encoding spore coat proteins of Bacillus subtilis.</section>
            ....
          </section>
        </section>
      </canonicalDocument>
  </acquisition>
```

Figure 11. *Exemple de document en entrée de la plate-forme Ogmios*

```
<documentCollection>
<documentRecord id="
     A79ACA58DEB7E6114747710B9A85059F">
  <acquisition>
...
  </acquisition>
  <linguisticAnalysis>
    <token_level>
    <token>
      <content>Combined</content>
      <from>0</from>
      <id>token1</id>
      <to>7</to>
      <type>alpha</type>
    </token>
    ...
    </token_level>
    <sentence_level>
    <sentence>
      <form>Combined action of two
           transcription factors
           regulates genes encoding
           spore coat proteins of
           Bacillus subtilis .</form>
      <id>sentence1</id>
      <refid_end_token>token30</
           refid_end_token>
      <refid_start_token>token1</
           refid_start_token>
    </sentence>
    ...
    </sentence_level>
    <semantic_unit_level>
    <semantic_unit>
      <named_entity>
        <form>Bacillus subtilis</form>
        <id>named_entity0</id>
        <list_refid_token>
          <refid_token>
            <refid_token>token27</
                 refid_token>
          </refid_token>
          <refid_token>
            <refid_token>token28</
                 refid_token>
          </refid_token>
          <refid_token>
            <refid_token>token29</
                 refid_token>
          </refid_token>
        </list_refid_token>
        <named_entity_type>species</
             named_entity_type>
      </named_entity>
    </semantic_unit>
    ...
    <semantic_unit_level>
    <word_level>
    <word>
      <form>Combined</form>
      <id>word1</id>
      <list_refid_token>
        <refid_token>
          <refid_token>token1</
               refid_token>
        </refid_token>
      </list_refid_token>
    </word>
    ...
    </word_level>
    <lemma_level>
    <lemma>
      <canonical_form>combined</
           canonical_form>
      <id>lemma1</id>
      <refid_word>word1</refid_word>
    </lemma>
    ...
    </lemma_level>
    <morphosyntactic_features_level>
    <morphosyntactic_features>
      <id>morphosyntactic_features1</id>
      <refid_word>word1</refid_word>
      <syntactic_category>JJ</
           syntactic_category>
    </morphosyntactic_features>
    <morphosyntactic_features>
      <id>morphosyntactic_features10</id
           >
      <refid_word>word10</refid_word>
      <syntactic_category>NN</
           syntactic_category>
    </morphosyntactic_features>
    ...
    </morphosyntactic_features_level>
  <syntactic_relation_level>
    <syntactic_relation>
      <id>syntrel1</id>
      <syntactic_relation_type>NCOMPby
      </syntactic_relation_type>
      <refid_head>
        <refid_word>word26</refid_word>
      </refid_head>
      <refid_modifier>
        <refid_word>word35</refid_word>
      </refid_modifier>
    </syntactic_relation>
    ...
  </syntactic_relation_level>
  </linguisticAnalysis>
 </documentRecord>
</documentCollection>
```

Figure 12. *Exemple de document en entrée et en sortie de la plate-forme Ogmios*